

David R. Jarema  
Edward H. Sussenguth

## IBM Data Communications: A Quarter Century of Evolution and Progress

*This paper describes the evolution of technological development in data communications at IBM. Rather than attempting to present a complete history, it emphasizes the changing environment and describes the more significant innovations that were incorporated in IBM's line of data communications products. Evolutionary developments in this area are traced from point-to-point batch transmission, to on-line batch communications, to interactive systems, and finally to networking. Although several aspects are treated, the primary focus of this account is on systems architecture, applications, and technology.*

### Introduction

Data processing has revolutionized man's capability to calculate and keep records. Advances in technology have provided the basis for the most significant societal change since the industrial revolution—the information society—which has been brought about by combining the dramatic technological achievements that have occurred in data processing with the rapid technological evolution of telecommunications.

In the last quarter century the application environment has changed from one where the telecommunication of data was basically a convenience (e.g., replacement of mail or couriers to transport input data and reports within an enterprise) to one where it has become an integral part of the day-to-day operation of business and government. In fact, there are many applications where communications provide the necessary backbone for implementation (e.g., airline reservations systems, missile command and control systems, information retrieval and management systems, automated bank teller systems).

In earlier days, business operations had to be modified in order to effectively utilize the available technology. Organizations were concerned with the relationship be-

tween business data flow and the computer system utilized at that time. In many cases, the technology defined how business could process its data. In today's world, technology provides the capability for a broad range of users, from very large international corporations to small independent businesses, to tailor the technology to their business needs rather than the converse.

This evolution has been due primarily to the significant changes offered by technology; not only has the price of data processing technology dropped dramatically, but system performance capability has shown a dramatic increase as well. While the cost of data communications has not decreased as dramatically, the utilization of new technology has contributed greatly to the stability of these costs in an inflationary environment.

Application growth and sophistication have placed heavy demands on technology, and the demand for continually improved cost/performance is obvious. Of equal importance is the increasing demand for significant improvements in system reliability and availability. When the base of comparison was the time it took to move data from one location to another by mail or by courier, any

Copyright 1981 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to republish other excerpts should be obtained from the Editor.

performance improvement from data transmission was welcomed, whereas in today's environment poor performance could have serious effects on the overall operation of the business.

In the last quarter century, while the ever-increasing use of data communications has revolutionized the data processing and information industries, the rate of development of telecommunications technology unique to data communications has been more limited. This is due primarily to the enormous existing base of telephone technology designed primarily for voice usage. Today, even with the increasing percentage of DP systems employing data communications, the amount of data traffic still represents a small portion of overall telecommunications usage, although this area is by far the most rapidly growing segment of telecommunications. Historically, data communications capability was provided by enhancing basic telephone or telegraph technology; today there are many new telecommunications networks being developed around the world which are designed exclusively for data transmission. These developments will undoubtedly lead to an even more explosive growth rate of data transmission in the years to come.

While it is difficult to chart the orderly evolution of a systems technology as multifaceted as data communications, the approach chosen here is to highlight the key elements of application and technology evolution. Data communications are traced from their beginnings in the early off-line point-to-point systems, through the evolution of batch, interactive, and job-oriented terminals and systems, to today's distributed processing environment. The evolution of systems architecture, networking, and software development are highlighted because of their critical contribution to the advancement of the overall technology and importance for the future.

#### **Point-to-point batch transmission of data**

The genesis of data communications can be traced to the first attempt to move data from a remote site into a computer center. From the very first, there were the problems of converting "languages" and of matching speeds between two dissimilar media, and these have remained important considerations throughout the history of data communications.

Prior to 1954, although most data were batch processed on punched-card-oriented equipment, telegraphic communication services were oriented toward paper tape. Therefore, a media incompatibility problem existed: Conversion from cards to paper tape was required at a remote site and from paper tape to cards at a central processing site. A significant innovation took place in 1940 when

IBM delivered a tape-to-card conversion machine to the U.S. Air Force. This machine made it possible to enter data from a remote site via teleprinter equipment, receive the data at the central site via paper tape, and then automatically convert to punched cards for processing. A year later, a card-to-paper-tape machine was developed for use at the remote sites and, for the first time, card-to-card transmission—without re-keying of the data a second time—was possible [1].

In order to eliminate the need for costly card-to-tape and tape-to-card conversions and to be able to transmit data more accurately and at higher speeds, the IBM Data Transceiver was developed and was announced in May 1954. This unit allowed direct card-to-card transmission and incorporated the following significant innovations:

- It was the first data terminal to use punched card media.
- It was the first commercially available data communications terminal that could operate over leased telephone circuits. It also operated over telegraph and radio channels.
- The serial card I/O equipment was limited in speed to 17 columns per second, which was significantly less than the transmission capacity of a voice-grade channel. The Data Transceiver was the first data terminal to effectively utilize the bandwidth available on a telephone circuit by partitioning the frequency spectrum into four subchannels (channelization). Through the use of plugable band-pass filters spaced 500 Hz apart, the user could operate up to four terminals simultaneously on a single voice-grade channel.
- It was the first data terminal to have a modem, integrated within the product, capable of operating over a leased telephone circuit.
- A unique 4-out-of-8-bit, fixed-count code was used in the Data Transceiver which permitted vertical checking since each code pattern transmitted had to have four *one* bits. (Telegraph transmission was based on the five-element Baudot code, which was unchecked for transmission accuracy.)
- With channelization of a leased voice circuit, up to 68 characters per second (cps) could be transmitted *versus* 6 to 10 cps on a telegraph channel.

With the expanded use of commercial computers in the mid-1950s, it became desirable not only to increase the speed of batch data terminals but also to transmit other forms of media such as magnetic tape. In 1960, IBM announced Synchronous Transmitter Receiver (STR), a new form of synchronous line control that was used in a new family of terminals.

Heretofore, with asynchronous transmission, additional bits had to be transmitted with each character to

define the character boundaries for synchronization. Maximum line efficiency is a function of the number of bits representing a character and the additional start/stop bits for each character. With STR, character synchronization was achieved by sending a synchronization pattern once per block of characters rather than for each individual character, thus increasing the line efficiency.

The first STR product transmitted from magnetic tape to magnetic tape. (It took seven hours to transmit a full reel of tape between two IBM 7701 tape drives!) Later versions used other media, permitting transmission in various combinations such as card to magnetic tape, card to card, paper tape to paper tape, card to computer, etc. Several specially designed STR devices were also successful, such as the IBM 1974, which combined an STR with a 1440 computer to become IBM's first programmed data communications product. STR operated over dial or leased telephone facilities. Like the Data Transceiver, STR was based on a 4-out-of-8 code that permitted transmission of up to 70 different code combinations. It also added a longitudinal parity check for improved accuracy, especially when using longer block lengths. The primary significance of STR was that it was the first broadly defined discipline for synchronous data communications between two devices. It not only permitted faster transmission over telephone facilities [1200 bits per second (bps) at time of announcement] but also improved line efficiency by using synchronous rather than asynchronous control. It also provided for error checking and introduced automatic retransmission of blocks containing errors.

For specialized applications, STR's speed was extended to operate at up to 5100 cps on 48-kHz broadband communications channels in 1962 and later up to speeds of 62 500 cps on <sup>®</sup>Telpak [2] D 960-kHz broadband channels.

In the late 1950s the concept of spooling was developed as a way of improving overall batch throughput in a computing system. Spooling permitted copying from relatively slow input media to an output medium (e.g., from magnetic tape to a printer) while a processor was performing other applications. With the introduction of System/360 and its operating system OS/360, all jobs to be processed were rerouted through a single job queue. Input readers and output writers were established as device-dependent programs to place work in a job queue and to send completed jobs to the appropriate output device. It was a logical extension of this concept to think of an input reader that used a telecommunications access method to gather input from data terminals for placement in the job queue, and an output writer to cause final out-

put to be transmitted to a terminal device. The OS/360 Remote Job Entry (RJE) program was announced in August 1967. With high-speed batch terminals containing a card reader for input and a printer for output, it became possible to submit jobs remotely to a System/360 computer using OS/360 RJE.

Until 1967, STR was the data link control used by all of IBM's high-speed batch terminals. The environment, however, was changing: The American Standard Code for Information Interchange (ASCII) was adopted as the standard interchange code by the American Standards Association [3], the System/360 was based on Extended Binary Coded Decimal Interchange Code (EBCDIC) 8-bit code, and users required the ability to transmit a bit stream that was transparent to the transmission system. Further, users desired to implement multipoint leased communications circuits in addition to point-to-point leased and dial facilities. Another key requirement was to have transmission control procedures which were code-independent; i.e., logic was built around the definition of transmission control functions without regard to their specific code implementation. Because of American National Standards Institute (ANSI) activities, transmission control functions were defined to be functionally identical to the ANSI definitions where possible. A superset of functions were defined where equivalent functions did not exist in the ANSI definitions. In order to satisfy these requirements a new byte-oriented line control known as Binary Synchronous Communications (BSC) was developed. BSC is a general-purpose data-link control that provides synchronous communications between a variety of terminals and transmission control units over either point-to-point or multipoint circuits. Some of the key control functions contained in BSC are as follows:

- Synchronization between the two devices;
- Establishment procedures—polling, selection, or contention;
- Circuit assurance—terminal identification;
- Message transfer—including delineation and blocking;
- Error checking—cyclic, polynomial checking or longitudinal checking, and automatic error recovery;
- Message acknowledgment;
- Termination procedures.

BSC supports IBM's EBCDIC code, ASCII code, and fully transparent text using EBCDIC controls. IBM developed BSC in parallel with national standards activities that led to the formulation of the ANSI X3.28 standard for data link control.

In January 1967, IBM announced the 2780 Remote Job Entry terminal that could transmit programs and data from a card reader at 400 cards per minute and print re-

ports from processed data received by the terminal at 240 lines per minute. It used BSC half-duplex line control, could operate at speeds up to 4800 bps on voice circuits, and was equipped with dual buffers so that transmission and card I/O and printing operations could be performed simultaneously.

#### **Job-specific and industry-specific terminals**

While early data base/data communications systems used general-purpose data terminals, it became evident that there were a number of applications that required unique terminals to meet the needs of specific growth areas.

One of the earliest applications identified was factory data collection in the manufacture and process industries. IBM's activities began in 1951 with a joint study with U.S. Steel Corporation. This study led to the announcement of the IBM 9600 Automatic Production Recording System in 1956. Transactions could be entered from terminal devices located throughout the factory floor, which were then transmitted via cable to a central console where the data could be punched into cards or printed. This system was followed by the IBM 357 Data Collection System announced in 1959. The 357 terminals permitted workers to record attendance or labor transactions by entering their badges containing punched holes, or pre-punched cards. The transaction data were recorded centrally onto punched cards. Both the IBM 9600 and IBM 357 were off-line systems and used multi-wire cable to connect the remote terminals to the central recorder.

The next IBM data collection system was the 1030, announced in May 1963. The 1030 system used standard two-wire communications facilities connecting remote terminal concentrators to the central computing system, making it the first on-line data-collection system. In addition to operating three times as fast as the 357, it permitted two-way response from the computer back to the workstation.

The finance industry was attracted to the use of on-line, data-communications-based systems to provide better service to savings deposit customers. IBM entered into a study with the First National Bank of Chicago in 1961 to define an on-line savings system, and the IBM 1060 System was announced in November 1962. The 1062 Teller Terminal had the following features:

- Passbook feeding and posting,
- Alphanumeric printing,
- A printed record of all transactions on a journal tape,
- Two-teller operation with separate net cash accumulators and locks, and
- Both on-line and off-line operation for increased reliability

In the late 1960s, a higher-function, higher-performance system, the IBM 2972/2980, was introduced. Key to this system was a new controller that could service up to ten teller terminals. The controller communicated with the central computer data base over 2400-bps voice circuits using BSC data link control.

It became evident in the early 1970s that advances in solid state technology and the rapidly increasing acceptance of on-line, data base/data communications systems made it attractive to pursue both other general-purpose and industry-specific applications. To provide the data terminal subsystems required for these new applications, IBM embarked on a program in 1970 to develop a family of "building blocks" that could be used in the multitude of data terminals and controllers required in the future.

The key to the building block concept was development of a series of upward-compatible microcomputers, known as "universal controllers" or UCs. The first UC appeared in the IBM 3600 Finance Communication System announced in August 1973. Since that time, six different UC models involving three generations of technology have been available as building blocks. The basic architectural philosophy of the UC microcomputer family includes the following: eight priority interrupt levels; one set of sixteen 32-bit registers per interrupt level; a direct-memory addressability to  $2^{16}$  bytes and a virtual addressing capability to  $2^{32}$  bytes; either direct control or channel control for I/O devices; and an instruction set of approximately 250 operations [4].

Today, the larger UCs, used in the 8100 family of distributed processors, contain more than a megabyte of main memory and operate at about 400 000 instructions per second. The smaller, single-chip UC, used primarily in terminal workstations and I/O adapters, contains a subset of the UC architecture, typically has 16K to 64K bytes of memory and performs about 150 000 instructions per second.

The key concept of the UC family is compatibility, both from programming and device-interface standpoints. Having a standard device-attachment interface has permitted IBM to develop a family of standard I/O and communications adapters as well as the I/O devices themselves. Today's data terminal developer now has a full range of "building block" components from which to configure his general-purpose or industry-specific product. These components include keyboards, printers, power supplies, diskettes, disk files, card readers and punches, CRT and gas-panel displays, as well as the standard adapters to provide an interface between these devices and the microcomputer. A range of standard com-

munications adapters are also available to the product developer.

An example of how this concept is helping to satisfy the requirements of unique industries is seen in the IBM 3660 Supermarket System. In this system, a UC-based in-store controller is equipped with electronic point-of-sale terminals that also contain UC microcomputers, displays, keyboards, and laser scanners. The purpose of the laser scanner is to automatically read the Universal Product Code (UPC) [5] imprinted on the label of each item. Knowing the code for each item, the system can automatically look up the price for that item and print it along with an alphabetic item description on the sales tape.

The UPC symbol is a distinctive line pattern that appears on all grocery items. The UPC pattern was developed by IBM and was submitted to the supermarket industry symbol selection committee. IBM also supported the industry-wide quantification of symbol print quality. The successful development of the UPC led to a proposal for a world product code, of which UPC would be a compatible subset. This proposal has had a favorable response in Western European countries as well as in Japan and Australia. Many countries have established national coding authorities and have instituted manufacturer marking of symbols on both food and non-food items.

The IBM 3687, a second-generation scanning system, was announced in November 1980. This system embodies the first commercial application of holographic technology in IBM. Through the use of a laser and a rotating holographic disk, a sophisticated scan pattern is generated which wraps around the UPC-coded item as it is scanned. This minimizes the need for the clerk to align the UPC symbol prior to scanning, thus improving the productivity of the checkout operation.

Modularity and scanning capabilities have also been extended to banking and to plant communications systems with the introduction of the IBM 3630 system.

### **Interactive systems**

Probably the earliest occurrence of a terminal operating interactively with a computer over communications facilities can be traced to a demonstration conducted by Bell Laboratories for the American Mathematical Society in September 1940 [6]. Bell had built a relay computer, which was installed in their laboratory in New York City. To demonstrate the computer to members of the Mathematical Society who were meeting in New Hampshire, they connected a teleprinter via a telegraph circuit to the New York City computer.

In early 1950, the U.S. Defense Department concluded that a new system needed to be developed to protect the U.S. from a manned bomber threat. A system called SAGE (Semi-Automatic Ground Environment system) was defined by M.I.T.'s Lincoln Laboratory [7]. IBM was awarded the contract to develop, manufacture, install, and maintain the central computer (AN/FSQ-7) and the combat control central (AN/FSQ-8). The AN/FSQ-7 included the communications front end that accepted information in real time via attached communications lines from tracking devices. Processed data were sent out over communications lines to defensive weapon systems. In a national network of Q-7s, each computer exchanged data with neighboring Q-7 computers and with a high-level Q-8. The significance of the SAGE effort was that it demonstrated the feasibility of merging a network of computers and communications to provide system solutions for complex, real-time applications [8].

In 1954, IBM established a joint research project with American Airlines to study their reservations problem and to define a communications-based system that would satisfy their requirements. A formal proposal was made and a contract was signed in November 1959. The major improvement in the proposed IBM system, known as Semi-Automatic Business-Related Environment (SABRE) [8], over other early electronic reservations systems, was that it handled not only the seat inventory but the passenger name records as well. The SABRE system also interconnected to other airlines via telegraphic facilities to automatically request space segments on their flights. The system, which became fully operational in 1964, consisted of duplexed processors, files, drums, and communications front ends. Over 1100 agent terminals located throughout the country were connected to the system by nine full-duplex leased voice circuits. The SABRE system, which capitalized on our SAGE experience, contained the significant innovations described in the following subsections.

#### **• Line control**

SABRE line control was developed for use on synchronous multipoint leased voice circuits. Hub-go-ahead polling was introduced. In this type of polling, the computer polled the furthest terminal, which then notified the next terminal when it could begin sending, and so on. The key advantage of this approach was the capability to poll or solicit traffic from a large number of interactive terminals on a multi-drop line with minimal overhead, providing an excellent response time. Classical polling techniques could not have met the response requirements established by American Airlines.

#### **• Remote multiplexers**

The SABRE system contained terminal interchange units

in 43 locations that interfaced the agents' workstations to the high-speed communications facility. When the agents' terminals and the terminal interchange unit were located in the same building, the terminals were cable-connected. Thus the terminal interchange unit acted like a clustered terminal controller. When the agents' terminals were remotely located from the terminal interchange unit, they were connected by low-speed communications circuits. In this case, the terminal interchange unit performed a remote multiplexing function.

- *Agents' terminal*

The agents' terminal was the first application of the IBM® SELECTRIC typewriter printing mechanism as a terminal print unit. The application opened up the use of this mechanism in many other IBM terminal products and in those of other manufacturers.

- *Real-time operating system*

SABRE had a requirement for a very efficient control program to handle the message throughput and provide the terminal response required.

- *Real-time front end*

The IBM 7286 Real Time Channel was developed for the SABRE system. This unit attached to the processor and had buffers so that a 36-bit word could be assembled from each communications line before the information was transferred to the processor unit.

- *Line testing project*

When the SABRE project was started, the maximum transmission rate was 1200 bps over leased voice circuits. Because of a very extensive line testing program IBM did in cooperation with AT&T, the rate for reliable transmission was raised to 2000 bps and later to 2400 bps on leased voice-grade channels. Improvements in signal processing have continued and now it is possible to transmit at speeds up to 9600 bps on an unconditioned voice channel using current modem products.

During the late 1950s, various government agencies and large corporations developed an interest in replacing their manual, torn-tape or semi-automatic message-switching centers with computer-controlled store-and-forward message-switching systems. The terminals used were largely telegraph keyboard printers or paper-tape readers and punches. While these terminals were adequate for administrative message traffic, they were functionally inadequate for data applications. Thus, in 1963, IBM announced the IBM 1050 Data Communication System for general-purpose applications. Some of the features and innovations of the 1050 include the following.

- *Line control*

Prior to this time, these systems used asynchronous,

start-stop line control with teletype line control in which the character set was limited to the 5-bit Baudot code. Bit and character synchronization was accomplished, character-by-character, by detecting the presence of a pulse at the start of a character and at the end of the character. Error detection was left to visual inspection of the received message. The IBM asynchronous line control permitted operation with both point-to-point and multi-point circuits. It used a new 7-bit, checkable code known as PTTC/BCD that permitted handling larger character sets than did the Baudot code. The new line control improved transmission efficiency by eliminating the FIGS and LTRS codes required in Baudot code. Both positive and negative acknowledgments were provided. Single control characters rather than control-character sequences were used, and the control structure permitted device selection at the receiving terminal.

- *Error retransmission*

The paper tape and card I/O were designed to back up and re-transmit the message in case an error was detected by the receiving unit.

- *Speed*

The terminal operated in half-duplex mode at 14.8 cps rather than at the maximum of 10 cps in teleprinter terminals.

- *Printer*

The printer used the IBM SELECTRIC typing element. This device provided superior print quality and lines wider than 78 characters.

In 1965, IBM introduced the 2740/41 communicating typewriter terminals. Great care was taken to ensure that all of the functional and human factor characteristics found in the standard IBM SELECTRIC typewriter were retained.

These devices, coupled with application software on the System/360, permitted a person to interact with a processor in a conversational manner. The SELECTRIC typewriter with its changeable typing element allowed it to be used for interactive computing applications such as APL, BASIC, and TSO, and also for business applications. Because of its high print quality, the 2741 was also used as the terminal in text-processing applications.

### **Display terminals**

The first IBM display terminal products introduced a new dimension to the man-machine interface; to the layman it was a window into the CPU, to the customer it offered new data entry capability with soft-copy printing, and to

those with vision, it was a more efficient linking of man and machine and one that possessed enormous potential for growth in customer interaction and productivity of computer systems. The key was developing the capability to enter, display, and manipulate data at terminals and the marriage of this capability, through communications facilities and discipline, with host programming to provide new and more efficient system applications.

The 2250 display family introduced in 1964 established capabilities that were to stand for years in high-function interactive vector graphics. The 2250 display presented an information content of up to 2800 characters or 6000 inches of vectors in a 1024 × 1024-dot addressable image space.

The history of alphanumeric terminal development was auspicious. The 2260/2848 product combined TV technology with delay lines for image refresh, to provide an information content of 240, 480, or 960 characters representing approximately 34 000 displayable points per terminal. By today's standards, terminal functions were low: erase-to-end-of-line, tab-to-column, and one shared subsystem printer. System communication was at 1200 and 2400 bps with start-stop protocol. This product laid the foundation for future inquiry and data entry applications and became the predecessor of the first display terminal for airline reservations.

The 3270 family introduced in 1972 established new standards in function, performance, and reliability that resulted in user acceptance far in excess of expectations. Key functions included the following:

- The 3270 data stream, formatted for CRT applications.
- 1920-character screen format of 24 rows by 80 characters, providing approximately 121 000 displayable points.
- Terminal functions such as character or line insertion or deletion, light pen, printer option, automatic-cursor-movement capability, data compaction, protected data, multi-level brightness, and signature recognition.
- BSC protocols with data rates to 7200 bps.
- Improved human factors serviceability and usability.
- Host DB/DC programming support (CICS/IMS, TCAM, VTAM).

In 1974 the plasma-gas-panel technology was introduced in a 480-character display for the 3600 Banking System.

The next generation of 3270 alphanumeric products emerged in 1977 and introduced new functional capabilities within a framework of improved quality, serviceability, and human factors oriented to enhance customer productivity. These included

- Larger screen formats: 43 rows of 80 characters and 27 rows of 132 characters, with an information content of approximately 360 000 displayable points.
- SDLC communication protocols with signaling rates up to 9600 bps.
- Enhanced character fonts.
- Extended data stream functions: multiple presentations of color, character graphics, text and alphanumeric data; growth of device independence at the terminal.

In 1978, introduction of the 3730 system marked the beginning of a text/word-processing application class of alphanumeric terminals.

The 8775 terminal shipped in 1979 introduced use of a microcomputer to offer new flexibility in function plus a loop-communications attachment to distributed systems. In the same year, the 3250, a follow-on product to the 2250, established new levels of cost/function in support of the sophisticated customer base that had developed over the years in the application of interactive design graphics.

In 1980, the 3270 family of alphanumeric products were further enhanced by two major new terminal- and host-supported functions:

- Low-cost presentation graphics using alphanumeric techniques which displayed over 371 000 points.
- High-resolution seven-color displays.

The consequence of these products is that displays have achieved the most prolific growth and dominant role of any interactive system terminal; it is a chronology of success.

#### **Network architecture and control**

As data communications networks became more complex, the interaction among their many parts became more critical. To control the flow of information between the terminal and the application program requires coordination among the terminal design characteristics, the application requirements, the operating system of the main computer systems, and the people who interact with the information (both users at the terminals and specialists coordinating the network operation). Among the key questions influencing this coordination are the following:

- Is direct message control or a queued system needed?
- Is the terminal to be associated with only one application or should an indefinite number of applications be accessible from the terminal?
- Do the programs and data all reside in one computer complex or are they distributed among several?
- What are the network size, performance, and reliability requirements?



The possibilities in each of these areas have evolved substantially since the 1950s when the first systems were being developed.

The development of general-purpose communications controllers was preceded by the introduction of special-purpose controllers. Special-purpose communications controllers of the 1950s were tightly coupled into systems such as the SAGE national defense system and the IBM 7281 Real-Time Channel, which was part of the Project Mercury man-in-space system. Other special products of the early 1960s, like the IBM 7286 Real-Time Channel, were designed for large, complex, and special applications and found use within large enterprises such as the Project MAC time-sharing experiments at M.I.T.'s Lincoln Laboratory and the passenger reservations system for American Airlines. These systems demonstrated the feasibility of interconnecting large networks in commercial applications.

The first general-purpose communications controller IBM developed was the IBM 7750 Programmed Transmission Control Unit, a stored-program front end that could attach to the IBM 1410/7000-series systems. Sometime later it was felt there was a need for a stand-alone communications processor for store-and-forward message-switching applications. The IBM 7740 was announced in June 1963 to serve this requirement. While it had lower line-handling capacity than the 7750, it had significantly greater throughput capability and attached disk files.

The introduction of System/360 in 1964 helped make teleprocessing cost-effective for a much wider scope of business enterprises. The IBM 2701, 2702, and 2703 Transmission Control Units were developed as general-purpose "hardwired" machines without programming capability and most of the sophisticated control function was relegated to the System/360 host computer.

#### **Access methods**

Another class of control functions, implemented with software and known as telecommunications access methods [9], were fundamentally of two types: direct control and queued. The Queued Telecommunications Access Method, QTAM, was developed in 1963. Its major design characteristics included compatibility with Operating System/360 data-management design and conventions; significant improvement in ease of use for large teleprocessing systems (at that time defined as systems in the range of 5-100 lines); and efficient use of low-speed communications lines (less than 600 bps), CPU time, and CPU storage. The three major innovations in the QTAM design were the concept of a message control program,

the use of a nonprocedural language (provided by OS/360 macros) to generate control program functions, and use of an allocating and scheduling mechanism.

Basic Telecommunications Access Method (BTAM), a lower-level access method than QTAM, was designed in 1964 to provide more immediate or direct control. Its major design characteristics included compatibility with OS/360 and DOS/360 data management design and conventions, use by OS QTAM, economy of CPU cycles and storage, and adaptability for use by higher-level packages and for incorporation of additional I/O devices (locally or remotely attached).

BTAM was used more widely than had been originally planned. It was used by various higher-level programs (*e.g.*, CICS and IMS), and it supported a large number of telecommunications devices and some nontelecommunications devices (*e.g.*, locally attached paper tape).

By 1966 it was realized that it would be very difficult to incorporate BSC support into QTAM because the mixture of the new higher data rates implied by BSC (up to 4800 bps) with the existing low data rates of start-stop protocols required a new buffer-management design. A design study completed in 1968 resulted in the introduction of TCAM (Telecommunications Access Method) as a replacement for QTAM. It was hoped at that time that TCAM would be a successful replacement not only for QTAM but also for BTAM.

This hope was not fulfilled because a direct control access method was still needed (TCAM provided a queued interface, as had QTAM), but BTAM needed substantial improvements to match the technology advances of System/370 virtual memory, Synchronous Data Link Control (SDLC), new programmable communications controllers, and the introduction of distributed processing. Virtual Telecommunications Access Method (VTAM) was developed in 1972 to meet these goals and to be the primary access method for IBM's overall telecommunications architecture plan, Systems Network Architecture (SNA).

#### **Communications controllers**

Before describing SNA, however, it is important to understand the contribution of the communications controller. Remote terminals interconnected with a centrally located computer by telephone lines require a device known as a communications controller at the central site to prepare, transmit, and receive the data for these terminals. The technology of the late 1960s established that stored-program concepts could be applied to small computers. In 1972 and 1973, IBM extended these concepts into communications controllers with the 3704 and 3705.



Almost a decade earlier the 7750 programmable communications controller had accommodated the complex teleprocessing control functions for large and special-purpose systems. Now the concept of moving many of these functions outboard of the System/370 into a communications controller was cost-effective for large and small system sizes.

The control program for the 3705, the Network Control Program (NCP), off-loads substantial processing from the System/370; it performs basic data communications functions such as routing data traffic towards its destination, managing and controlling the flow of data traffic to avoid congestion, buffering and queuing of data, scheduling transmission line operations (*e.g.*, sending data and polling), performing various line control protocols (*e.g.*, BSC and SDLC) and error-recovery procedures, and detecting and reporting to its access method irrecoverable errors and other exception conditions.

In addition to introducing improved methods for these functions, the control program also is capable of emulating the IBM 270X controllers so that applications previously written with the BTAM access method will run concurrently with NCP and with VTAM- or TCAM-based applications. As teleprocessing requirements grew in the 1970s, these basic communications functions expanded into sophisticated networking capabilities managed by NCP in the 3705 and VTAM or TCAM in System/370, coordinated by SNA.

### **Systems Network Architecture (SNA)**

SNA significantly advanced the state of the art in teleprocessing software systems [10, 11]. Announced in September 1974 as IBM's blueprint for future teleprocessing products, SNA provides a unified design for the functions and structure of IBM's data communications products. It allows customers to establish a long-term direction as they plan networks of terminals and processors, both large and small. In this sense SNA is analogous to System/360, introduced in 1964 with its uniform architecture for processors, which was the response to the earlier problems of sharing programs, I/O devices, and other resources of the central computer complex among the variety of IBM 1400- and 7000-series computers. Prior to the introduction of SNA, teleprocessing networks were beset by many problems: Terminals were often dedicated to the use of a single application; numerous and diverse line-control procedures and terminal types were ingrained into the support programs, application programs, and network operations; and multiple access methods were in common use, thwarting any attempt to share resources among applications. Each of these problems made it difficult to expand existing applications or to add new ones. SNA was

introduced to solve these problems and to make teleprocessing applications easier to install, operate, and expand.

SNA also had its roots in the hardware technological advances of the early 1970s. At that time, it became economically possible to incorporate a small processor into the design of many terminals (such as the universal controller described previously).

Prior to such microcomputers, a terminal was commanded directly by its host computer. For example, each keystroke produced an input character transmitted independently at the rate of generation; and each output character was sent at a rate not exceeding that of the printer. Commands such as "shift to upper case" or "carriage return" were handled by the host on an individual basis. Thus, the host computer was intimately associated with the detailed characteristics of each device attached to it; and, as the terminal network expanded, the workload placed upon it could become excessive.

With the new microcomputer-based designs, the processor within the terminal handles many functions independently of the host, and the transmissions between host and terminal are complete messages sent at high speed. This reduces the processing power required at the host and/or allows more terminals for a host of the same size. A more important change, however, was in system structure. No longer is a tight coupling between terminal and host needed; device control now can be placed at or near the end terminal and not in the host. Thus, system commands, protocols, and procedures designed for tight coupling are no longer required; instead, a new set specifically designed for distributed processing is required.

Just as the processor in the terminal now handles device control, it also readily becomes an application processor. A simple example is found in a retail store: The point-of-sale terminal and an in-store controller combine to off-load the host by totaling individual purchases, computing tax and change, processing credit verification, maintaining inventory files, and preparing summary management reports. Again, the important point, from a system standpoint, is that the application may now be performed in any of several places within a network—at the host, at a controller, or even at the terminal itself.

This is a new structure that essentially did not exist before, and definitions for the control of such a system are needed. The advent of distributed processing, then, whether for device control or distributed application processing, was the fundamental technical rationale behind the creation of SNA.

From an architectural point of view, SNA is a top-down structured design composed of layers [12, 13]. The lowest layer, *data link control*, directly manages physical resources—the transmission facilities that connect nodes. Successive layers provide additional services. For example, the *path control* layer provides a routing service, so that its users are unaware of the physical topology of the network, and some nodes contain a *control point* that controls the nodes (e.g., terminals and controllers) and lines in their own portions of the network. Other layers provide services to applications; these can include transparent access to local or remote resources (e.g., DLI VSAM files), mapping of data streams to and from application data structures (also called presentation services), access to other local or remote programs, management of buffer commitments, and encryption of data before transmission and decryption upon receipt.

This layered view of a communications architecture was used in a network sponsored by the U.S. Advanced Research Projects Agency (ARPANET) in the late 1960s. SNA was the first commercial implementation of the layered concept and has been followed by most other computer manufacturers who offer teleprocessing software support and by the International Standards Organization (ISO) in its recommendations for system interconnection [14, 15].

As was later done in public packet-switching networks, SNA adopted the technique of sharing the use of links by time-division multiplexing on a message basis. Each message is switched at intermediate nodes to the next link along its route.

### Networking

The term *networking* is used to connote the concept of a geographical distribution of terminals (usually hundreds or thousands in ten to one hundred locations) working with application programs in computer complexes (usually one to ten or more computers in a like number of locations). The pioneering networks, such as for airlines and banks, were devoted to a single major application, such as reservations or savings deposit accounting, and constrained the variety of attached terminals usually to one type of display and one type of printer. Such restrictions have been largely eliminated in most networks in operation and being planned today. Now many application types, each with varying load and performance requirements, can operate concurrently; many varieties of displays, printers, and industry-oriented terminals can coexist in the same network; and each terminal no longer need be dedicated to a single use. Rather, it can be shared among applications varying from simple inquire-and-update to text processing and problem solving.

To control the single-application networks, the telecommunications access methods were tightly coupled with the application and, when tuned for performance and reliability, reasonable tradeoffs could be made between the technical requirements and the applications. These tradeoffs often took the form of unique link controls, specialized front-end processors, and extensive modifications of the telecommunications access methods. Many of these, such as the examples already mentioned, introduced new concepts, but for single rather than for multiple applications.

As the need for more general-purpose networking capability grew, so also did the need for codification of conceptual design so that hardware and software products would work in harmony, and so that each installation could be readily tuned for performance and reliability. The SNA plan, implemented in such System/370 software products as VTAM, TCAM, and NCP, now provides extensive leading-edge networking capability [16–18].

### ● Networking services

The initial releases of SNA (through 1977) provided user services for single-host tree configurations. A primary service was the sharing of network resources; for example, multiple terminals of varying types on a single SDLC link could interact with any of multiple applications in one host. This sharing is key to reducing network costs and providing flexible services.

The major objectives in the development of SDLC with respect to previous link controls included bit transparency, full-duplex transmission, improved error checking, capability of operation over long-delay high-speed facilities, and reduced protocol overhead [19–21]. IBM developed SDLC in parallel with international standards activities that led to the formulation of the ISO HDLC (High-level Data Link Control) standard. The use of a pause-and-retry error recovery algorithm in NCP also improved availability by allowing links to automatically recover from transient link problems without requiring operator intervention. Configuration-management services were provided by VTAM and TCAM, through which the network operator could activate and deactivate links and nodes, and receive notification of failures. Other services provided the connection between applications in the System/370 and those in the terminals or controllers.

A subsequent release of SNA in 1978 introduced multiple-host networks [22]. This included capabilities in which a terminal controlled by one host could access an application in any host in the network, and host-to-host sessions could also be established. The single control point (for session establishment and configuration ser-

vices) and hierarchical control were generalized to a network of multiple control points which operate as peers of one another. Further enhancements provided functions such as parallel links, transmission priority, and multiple active routes for data transmission. Parallel links may be used between adjacent nodes of a network to provide additional bandwidth and backup, and these parallel links may be logically grouped to automatically distribute traffic across the links of a group. This concept also compensates for degradation resulting from errors on any of the links in the group, because transmission is disrupted only if the last remaining link in the group fails. Network availability can also be increased by providing multiple routes between the same two points in a network, so that traffic can be rerouted (and disrupted sessions reconnected) to avoid failing intermediate nodes or failing links. Multiple routes can also be useful for traffic load leveling. These capabilities gave SNA the complete configuration flexibility of mesh networks, as distinct from the former tree structures and connection of trees.

To save costs, networks are normally designed so that the peak rate of traffic into the network may, at times, exceed the maximum network throughput. Queues in the network help smooth the peak loads, but flow-control mechanisms are necessary to prevent substantial throughput degradation, or even deadlocks, as the offered load increases beyond the network's capacity [23, 24]. *Flow control* operates by limiting the rate at which traffic is accepted into the network. SNA products use a flow-control mechanism based on *pacing* which allows a specific number of message units to be sent from one end of a route, after a pacing response is received from the other end. This number is dynamically adjusted by checking queue depths at the nodes along the route. The dynamically adjusted pacing values provide greater throughput than statically defined values used in other systems. Another aspect of SNA flow control is the use of message priorities, such that at each trunk line messages are transmitted in the order of the priority given to their respective sessions. Performance analysis has shown that in networks experiencing high loads, predefined routing with pacing also may provide better performance than routing schemes that allow routing decisions on every message without establishing explicit physical routes for session traffic.

The origins of many of these networking services may be found in older network designs, both research and commercial, but IBM improved many of the algorithms and was the first to offer such extensive total function to its customers.

#### • *Special services*

In addition to the fundamental network control and net-

work services offered, many other essential and innovative features are included; three of these are network testing and diagnosis, cryptography, and distributed data base sharing.

In 1979, an innovative series of modems was introduced. Not only are the traditional modem functions enhanced (for example, the IBM 3865 modem is the first to operate at 9600 bits per second on unconditioned lines and has an equalization time on multipoint lines about five times faster than other contemporary modems) but, in a unique way, modem testing and diagnostic capabilities were fully integrated into the network problem-determination facilities [25-28]. The tests intersperse diagnostic sequences with user data without disrupting ongoing sessions. Information provided by these modems to NCP and other programs can help to pinpoint errors in data terminal or communications facilities, or to identify unfavorable line quality trends. An operator using these facilities can initiate tests of modem pairs and the communications line between them.

In the early 1970s, scientists and engineers who were engaged in the study of solutions to security problems associated with data processing concluded that cryptographic techniques were required to secure information communicated through physically unprotected networks. In 1972, IBM researchers completed their work in the development of powerful cryptographic algorithms which were simple enough to be implemented on a single chip and yet secure enough for commercial applications [29-32]. This effort led to the development of the Data Encryption Standard Algorithm, which in 1977 was adopted by the National Bureau of Standards as the U.S. Federal Standard for Cryptography for data processing applications [33].

The system design aspects of cryptography proved to be particularly challenging. "End-point-to-end-point" encryption [30] is necessary; that is, in most instances encryption/decryption operations should be performed by the sender and receiver of the information; intermediate nodes, through which the information may flow, should not have access to non-encrypted data. The SNA architecture proved particularly helpful in meeting these objectives, since its protocols and formats provide easily partitioned headers (which are not encrypted), and data fields (which are encrypted) that are unconstrained from any coding considerations. Other system design objectives included the following: Each data set or record may be encrypted under a different key and each node may have a different cryptographic key; the cryptographic key must not be subject to manipulation by the control program, even if the control program is modified by unauthorized

personnel; and through the use of authorized hierarchical keys, files encrypted by one system can be read (and written) by any other authorized system without the necessity of sharing knowledge of the keys used for enciphering the data. In order to accomplish these objectives, a new hierarchical key-management scheme was devised that permits the use of system facilities without compromising the security of the encrypted system-key tables [30].

As network components become more intelligent and sophisticated, distributed processing and distributed data bases will become increasingly commonplace. Processors introduced in the late 1970s such as the 4300, 8100, Series/1, System/34, and System/38 emphasize this trend. Applications are being developed that are comprised of programs residing in several processors in several locations. These geographically dispersed programs, in turn, use data files distributed in several locations, often different from the locations of the programs. An example of such a system might be inventory control in which parts-availability information is located at the production site, parts-on-order information is located at various receiving docks, vendor-billing data are processed at one central computer complex, and vendor-quality-control statistics are analyzed at another. To facilitate the coordination, error recovery, and efficient operation of such systems, SNA products include conventions for program-to-program operation in addition to terminal-to-program operation. The intersystem communications capability [13, 34] of Information Management System (IMS), Customer Information Control System (CICS), and Distributed Processing Programming Executive (DPPX), announced in 1979, provides the capability of each of these products to communicate with one another, to route messages based on content, and to support distributed data bases with the actual geographic location of the data masked from the using application programs.

#### **Internal IBM networks**

The IBM Corporation is not only a producer of data communications programs and products, but is also a major user of these products in support of its worldwide business operations. In the past ten years, internal use of data communications has grown to the point where development, manufacturing, marketing, field engineering service, and headquarters operations cannot function smoothly unless the internal networks and associated applications are available to IBM personnel worldwide. In the 1960s, most operating units met their needs with dedicated networks. In the 1970s many of these dedicated networks were merged for reasons of economy and efficiency.

Five very large international networks emerged from this consolidation. It is instructive to look at just one of

these, the Corporate Consolidated Data Network (CCDN), as it is a good example of usage and growth typical in large industrial corporations. The other networks [35] are similar but are operated independently of CCDN because of their size (*e.g.*, the System Communications Division's network has 8000 terminals), or their unique geographic orientation (*e.g.*, for Europe, the Middle East and Asia) or their operating characteristics (*e.g.*, international bulk transmission).

The CCDN network is a centrally managed, high-performance interactive network that supports both SNA and older applications. All 15 000 terminals are SNA terminals. The System/370 concentrators are located in four cities across the United States. They are unattended and are remotely managed by the System/370 Network Control System in New York. The eighteen host application sites are geographically dispersed throughout the United States.

The growth of CCDN usage, which has been approximately 25% per year, can be appreciated by reviewing a few statistics. Let us compare the years 1975, 1978, and 1981 for several important parameters: The number of terminals has increased from 4400 to 8100 to 15 400; application programs, from 3 to 14 to 30; and characters transmitted per year, from 75 to 204 to 460 billion. The applications available to the 40 000 CCDN users include order entry, financial analysis, proposal preparation, message switching, education and training, and history of repair actions.

#### **Satellite transmission**

In 1965, the Intelsat consortium successfully launched "Early Bird," the first geostationary communications satellite for telephone and television transoceanic service. Shortly thereafter, it was recognized that communications satellites were significant for future data processing systems, and IBM initiated studies that led to a time-division multiple-access (TDMA) system to efficiently serve both voice and data systems. By 1973, a precursor IBM system known as Telesatcom, using the Canadian Anik Satellite, had implemented voice, data, and image communications using TDMA. In another test, earth stations in Poughkeepsie, New York, and Rochester, Minnesota, efficiently communicated at 1.544 Mbps.

More extensive testing in 1977 and 1978 between sites in the United States, France, and Germany using the European Symphonie satellite, also at 1.544 Mbps, demonstrated the feasibility of distributed data processing functions such as remote job entry, tape-to-tape and disk-to-disk file transfer, and dynamic-network load sharing [36]. The success of the Telesatcom precursor led to the instal-

lation of an operational system in the United States, beginning in 1978, providing IBM employees in 29 cities with telephone communications via satellite. The Telesatcom experience has influenced the development of a highly sophisticated satellite communications controller for Satellite Business Systems, which efficiently compresses and merges voice and data streams and is the primary system component co-located with earth stations residing directly on customer premises.

### Special data communications applications

Finally, it might be instructive to take a look at the three special communications applications outside of the main stream of data communications development.

- *Audio response*

Work in audio response can be traced back to early development in Endicott in 1954. A device called the "VOX-BOX" had a vocabulary of twelve words on a magnetic drum. The device could answer a telephone call and then respond with audio messages based on the digits dialed.

In 1961, a project known as Voice Answer Back (VAB) was initiated in Poughkeepsie. The VAB unit stored words or phrases on tracks of a magnetic drum. Techniques were developed for digitizing the voice recording and expanding it or compressing it as required so that it occupied a full revolution of the drum when converted back to analog form. IBM's first product, the IBM 7770 Audio Response Unit, was announced in January 1964. A precursor unit was installed in the New York Stock Exchange quotation system. IBM was the first company to introduce a commercially available standard audio response product.

The significance of this innovation was that it

- Added a new form of terminal output—the spoken word.
- Made the standard \*Touch-Tone [2] telephone a terminal for inquiry-response applications.
- Opened up a number of significant new applications, heretofore not economically feasible, such as credit verification, bank-balance inquiry, and telephone intercept.

- *Telephone intercept*

In 1963, work began on the development of a system for telephone companies that could automatically intercept telephone calls to non-working numbers. By looking up information stored in a computer data base, the IBM 7770 Audio Response Unit could then transmit a message such as "the number you have called has been changed to xxx-xxxx." The 2910 Automatic Intercept System [37] was connected directly to telephone central-office switching

equipment. The first system was installed at Southwestern Bell in St. Louis, Missouri in January 1965.

- *Stored-program-controlled PABX*

Telephone exchanges up to the early 1960s were almost solely electromechanical. Since then semi- or fully electronic exchanges have been developed and the control portion of these switching systems uses stored-program computers.

IBM began a development effort on a stored-program private automatic branch exchange in the mid-1960s. In May 1969 it announced the IBM 2750 Voice and Data Switching System for marketing in five European countries. The principal features of the IBM 2750 were

- A space-division switch using solid state crosspoints for the first time;
- Stored-program control, supporting both voice and data applications;
- A BSC adapter permitting communications with a System/360;
- High reliability and availability through redundant design;
- Attachment of a wide variety of terminal devices in addition to telephone handsets.

In 1972 a follow-on system, the IBM 3750, was announced. The 3750 can handle up to 2500 lines, has networking capabilities, has greater traffic-handling capacity than the 2750, and better reliability and diagnostic characteristics. The 1750, announced in 1979, handles from 100 to 760 lines.

### Conclusion

The last twenty-five years have seen significant changes in data communications. At first, it seemed that only unusual applications required data to be moved rapidly from one place to another. Today, we have airline reservations systems, banking systems, and data communications networks with terminals in stores, offices, and factories. In the U.S. work force in 1980, there is a terminal for every 48 employees. Among IBM customers there are 25 employees per terminal. And within IBM itself we have about five employees per terminal. By the mid-1980s, it is estimated there will be a terminal for every ten employees in the U.S. work force: one for every six of our customers' employees, and one for every two IBM employees [38].

The changes in technology have been as spectacular as the growth in numbers. Now we have interactive as well as batch processing; shared and dedicated lines; and high-speed, synchronous as well as slow-speed, start-stop data transmission. Many uses remain for keyboard input and printer output, but laser scanners and visual displays, and

even audio response, make it easier for people to use data communications systems. Hardwired, inflexible terminals are being replaced by distributed processing with programmable terminals.

In the remaining two decades of the century, convenient terminals equipped with display screens—not requiring special skills for operation—should move into many offices, schools, and then into homes. Distributed data bases will allow people ready access to information and special services of many types. Data, voice, and facsimile services will probably become available from a single all-digital system.

### Acknowledgments

The authors wish to express their appreciation to those too numerous to mention who contributed to the writing of this paper. More importantly, recognition is also due to the hundreds of colleagues who brought to fruition the many successful IBM telecommunications products.

### References and notes

1. C. R. Doty, Sr., "Data Communications," *Technical Report TIC 64G7-4429*, IBM Data Systems Division, Poughkeepsie, NY, July 1, 1964.
2. ®Telpak and ®Touch-Tone are registered trademarks of AT&T.
3. This organization is now known as the American National Standards Institute.
4. *IBM 8100 Information System, Principles of Operation*, Order No. GA23-0031, available through IBM branch offices.
5. D. Savir and G. J. Laurer, "The Characteristics and Decodability of the Universal Product Code," *IBM Syst. J.* **14**, 16-34 (1975).
6. E. G. Andrews, "Telephone Switching and the Early Bell Laboratory Computers," *Bell Syst. Tech. J.* **42**, 341-353 (1963).
7. R. R. Everett, C. A. Zraket, and H. D. Benington, "SAGE—A Data Processing System for Air Defense," *Proceedings of the Eastern Joint Computer Conference*, Washington, DC, 1957, pp. 148-155.
8. Gilbert Burck and the Editors of Fortune, *The Computer Age*, Harper and Row, 1965.
9. D. L. Mills, "Communication Software," *Proc. IEEE* **60**, 1333-1341 (1972).
10. E. H. Sussenguth, "Systems Network Architecture: A Perspective," *Conference Proceedings*, 1978 International Conference on Computer Communications, Kyoto, Japan, 1978, pp. 353-358.
11. D. Doll, "IBM Strengthens its Architecture," *Data Communications* **8**, 56-67 (1979).
12. R. J. Cypser, *Communications Architecture for Distributed Systems*, Addison-Wesley Publishing Co., Reading, MA, 1978.
13. *SNA Technical Overview*, Order No. GC30-3073, available through IBM branch offices.
14. F. P. Corr and D. H. Neal, "SNA and Emerging International Standards," *IBM Syst. J.* **18**, 244-262 (1979).
15. *Reference Model of Open Systems Architecture*, Document ISO/TC97/SC16/N227, International Standards Organization, 1979; available from American National Standards Institute, Inc., 1430 Broadway, New York, NY 10018.
16. S. Scott, "VTAM Means Software for More Logical Network Management," *Data Communications* **8**, No. 1, 77-90 (1979).
17. L. Esau, "How to Access a Network via IBM's TCAM," *Data Communications* **8**, No. 2, 89-106 (1979).
18. A. Hedeem, "Networking: Building a Software Bridge Between Multiple Hosts," *Data Communications* **8**, No. 3, 87-100 (1979).
19. J. R. Kersey, "Synchronous Data Link Control," *Data Communications* **3**, No. 5, 49-60 (1974).
20. R. A. Donnan and J. R. Kersey, "Synchronous Data Link Control: A Perspective," *IBM Syst. J.* **13**, 140-162 (1974).
21. *Synchronous Data Link Control General Information*, Order No. GA27-3093, available through IBM branch offices.
22. J. P. Gray and T. B. McNeill, "SNA Multiple-System Networking," *IBM Syst. J.* **18**, 263-297 (1979).
23. G. A. Deaton and D. J. Franse, "A Computer Network Flow Control Study," *Conference Proceedings*, 1978 International Conference on Computer Communications, Kyoto, Japan, 1978, pp. 135-140.
24. V. Ahuja, "Routing and Flow Control in Systems Network Architecture," *IBM Syst. J.* **18**, 298-314 (1979).
25. R. A. Weingarten, "An Integrated Approach to Centralized Communications Network Management," *IBM Syst. J.* **18**, 484-506 (1979).
26. Dominique Godard and Daniel Pilost, "A 2400-Bit/s Microprocessor-Based Modem," *IBM J. Res. Develop.* **25**, 17-24 (1981).
27. Simon Huon and Robert Smith, "Network Problem-Determination Aids in Microprocessor-Based Modems," *IBM J. Res. Develop.* **25**, 3-16 (1981).
28. J. J. Budway, "Software Tackles the Task of SNA Net Control," *Data Communications* **8**, No. 5, 83-93 (1979).
29. J. L. Smith, W. A. Notz, and P. R. Osseck, "An Experimental Application of Cryptography to Remotely Accessed Data System," *Proceedings of the ACM Annual Conference*, Boston, MA, August 1972, pp. 282-297.
30. W. F. Ehrsam, S. M. Matyas, C. H. Meyer, and W. L. Tuchman, "A Cryptographic Key Management Scheme for Implementing the Data Encryption Standard," *IBM Syst. J.* **17**, 106-125 (1978).
31. S. M. Matyas and C. H. Meyer, "Generation, Distribution, and Installation of Cryptographic Keys," *IBM Syst. J.* **17**, 126-137 (1978).
32. R. E. Lennon, "Cryptography Architecture for Information Security," *IBM Syst. J.* **17**, 138-150 (1978).
33. *Data Encryption Standard, Federal Information Processing Standard Publication 46*, National Bureau of Standards, U.S. Department of Commerce, Washington, DC, 1977.
34. *CICS System Application Design Guide*, Order No. SC33-0068, available through IBM branch offices.
35. R. S. Moore, "Evolution of a Laboratory Communication Network," *IBM Syst. J.* **18**, 315-332 (1979).
36. *Computer/Satellite Communications Experiment; Phase 2 Final Report*, Joint IBM/Comsat/French PTT/DFVLR Report, submitted to Federal Communications Commission, 1919 M St. NW, Washington, DC 20554.
37. G. F. Abbott, R. L. Bence, J. A. Ceonzo, and J. M. Regan, "Design of an Automatic Telephone Intercept Switch," *IBM J. Res. Develop.* **9**, 274-281 (1965).
38. L. M. Branscomb, "Computer/Telecommunications Technology in the 80s," presented at the International Communications Association Annual Meeting, Detroit, MI, May 13, 1980.

Received April 18, 1980; revised September 19, 1980

The authors are located at the IBM System Communications Division laboratory, Research Triangle Park, North Carolina 27709.